

Devang Patel Institute of Advance Technology and Research



Department of Computer Engineering CE264 Design & Analysis of Algorithms

Practical – 1

Aim: Implement and analyse algorithms given below:

- 1.1) Factorial (Iterative and Recursive)
- 1.2) Fibonacci Series(Iterative and Recursive)
- 1.3) GCD (Iterative and Recursive)

Program Code:

```
import java.util.*;
public class master
  public static int factorial(int n)//recursive factorial
     if(n==1)
        return 1;
     return (n*factorial(n-1));
  public static int fact(int n)//iterative factorial
     int fact=1;
     for(int i=1 ; i \le n ; i++)
        fact=fact*i;
     return fact;
  }
```



Devang Patel Institute of Advance Technology and Research



Department of Computer Engineering CE264 Design & Analysis of Algorithms

```
public static int fib(int n)//recursive fibonacci
  if(n==0 || n==1)
     return n;
  return (fib(n-1)+fib(n-2));
}
public static int ffib(int n)//iterative fibonacci
  int pprev, prev = 0, curr = 1;
  for (int i = 1; i < n; i++)
     pprev = prev;
     prev = curr;
     curr = pprev + prev;
  return curr;
public static void main(String args[])
  long begin_time=System.currentTimeMillis();
  System.out.println("factorial of 5= "+factorial(5));
  long end_time=System.currentTimeMillis();
  System.out.println("time takenby recursive approach= "+(end_time-begin_time));
```



Devang Patel Institute of Advance Technology and Research



Department of Computer Engineering CE264 Design & Analysis of Algorithms

```
long ibegin_time=System.currentTimeMillis();
     System.out.println("factorial of 5= "+fact(5));
     long iend_time=System.currentTimeMillis();
     System.out.println("time takenby iterative approach= "+(iend_time-ibegin_time));
     long fbegin_time=System.currentTimeMillis();
     System.out.println("fibonacci series "+fib(7));
     long fend_time=System.currentTimeMillis();
     System.out.println("time taken by recursive approach= "+(fend_time-fbegin_time));
     long tbegin_time=System.currentTimeMillis();
     System.out.println("fibonacci series upto 5= "+ffib(6));
     long tend_time=System.currentTimeMillis();
     System.out.println("time taken by iterative approach= "+(tend_time-tbegin_time));
import java.util.*;
public class k
  //Iterative GCD
  public static int gcd(int a, int b)
  int result = Math.min(a, b);
   while (result > 0) {
     if (a % result == 0 \&\& b \% result == 0) {
        break:
```

}



Devang Patel Institute of Advance Technology and Research



Department of Computer Engineering CE264 Design & Analysis of Algorithms

```
result--;
  return result;
//Recursive GCD
public static int gcd2(int a, int b)
  if (a == 0)
     return b;
  if (b == 0)
     return a;
  if (a == b)
     return a;
  if (a > b)
     return gcd(a - b, b);
  return gcd(a, b - a);
public static void main(String args[])
  long begin_time=System.currentTimeMillis();
  System.out.println("GCD Answer= "+gcd(5,15));
  long end_time=System.currentTimeMillis();
  System.out.println("time taken by GCD= "+(end_time-begin_time));
  long ibegin_time=System.currentTimeMillis();
  System.out.println("GCD Answer= "+gcd2(5,15));
  long iend_time=System.currentTimeMillis();
```



}

Charotar University of Science and Technology

Devang Patel Institute of Advance Technology and Research



Department of Computer Engineering CE264 Design & Analysis of Algorithms

System.out.println("time taken by GCD= "+(iend_time-ibegin_time));
}

OUTPUT:

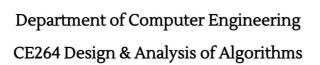
| | Sr.No. | Iterative(in seconds) | Recursive(in seconds) |
|-----------|--------|-----------------------|-----------------------|
| Factorial | 1 | 0.014 | 0.014 |
| | 2 | 0.017 | 0.021 |
| | 3 | 0.014 | 0.013 |
| | 4 | 0.015 | 0.015 |
| | 5 | 0.019 | 0.019 |

| | Sr.No. | Iterative(in seconds) | Recursive(in seconds) |
|-----------|--------|-----------------------|-----------------------|
| Fibonacci | 1 | 0.013 | 0.012 |
| | 2 | 0.017 | 0.014 |
| | 3 | 0.0112 | 0.028 |
| | 4 | 0.215 | 0.567 |
| | 5 | 0.013 | 0.105 |

| | Sr.No. | Iterative(in seconds) | Recursive(in seconds) |
|-----|--------|-----------------------|-----------------------|
| GCD | 1 | 0.014 | 0.024 |
| | 2 | 0.015 | 0.021 |
| | 3 | 0.014 | 0.015 |
| | 4 | 0.014 | 0.029 |
| | 5 | 0.019 | 0.026 |
| | | | |



Devang Patel Institute of Advance Technology and Research





CONCLUSION: For factorial function both iterative and recursive approach behaves same. For Fibonacci series iterative approach is better because as n increases recursive algorithm has exponential increase in time complexity. For GCD iterative approach is better and preferrable because as values increases recursive approach is inefficient and takes more elapsed time for computation.

Staff Signature:

Grade:

Remarks by the Staff: